Work Package 1, D1.3

Project Name:	Disc4All 955735		
Date:	30/ 04 / 2024	**Release:**	Final_v2
WP Leader (WPL):	OULU		
WP Co-leader (WPCL):	InSilicoTrials		
Reviewer 1	Marc-Antonio Bisotti (InSilicoTrials)		
Reviewer 2	Jerome Noailly		
Document Number:	D 1.3		

Revision History

Date of next revision:

Revision Date	Previous Revision Date	Summary of Changes	Changes Marked
22.12.2022	N / A	Technical Interoperability catered for but data has not been described adequately	Yes
26/01/2023	22/12/2022	Data described adequately and syntactic interoperability explained but semantic interoperability still in question	Yes
16/4/2024	03/04/ 2024	Semantic layer added & Secondary Data	Yes

Approvals

This document requires the following approvals. A signed copy should be placed in the project files.

Name	Signature	Title	Date of Issue	Version
Marc-Antonio Bisotti	YES	PI	1 / 26 / 2023	Draft2
Jerome Noailly	YES	PI, Coordinator	03/04/2024	Final v1
Jerome Noailly	YES	PI, Coordinator	30/04	Final_v2

Distribution

This document has been distributed to:

Name	Title	Date of Issue	Version
Marc-Antonio Bisotti	PI	22.12.2022	Draft1
Consortium	PIs, ESRs	03/04/2024	Final_v1
Consortium	PIs, ESRs	30/04/2024	Final_v2

Work Package Authorization	
Title	PI, ESR Supervisor
Person Authorized[1]	Marc-Antonio Bisotti
Date[2]	16.04.2024

Description & Deliverables

(A description of the work to be done)

Techniques, Processes and Procedures

(Any techniques, tools, standards, processes or procedures to be used in the creation of the specialist products)

Problem Handling and Escalation

(This refers to the procedure for raising issues and risks)

Approval method

(The person, role or group who will approve the completed products within the Work Package, and how the coordinator is to be advised of completion of the deliverables and Work Package)

Work Package Acceptance	
Person Accepting[3]	Jerome Noailly
Date[4]	30/04/2024

1 The name of the WPL
2 The date of the agreement between the Coordinator and the WPL/person authorised
3 The Coordinator or other person accepting the work package on the Coordinator's behalf
4 The date of acceptance

Assessment and feedback	

D 1.3 Primary and Secondary Data Interoperability

Abstract

The Disc4All consortium's research on intervertebral disc degeneration generates and utilizes a substantial amount of primary and secondary data. Data reusability, repurposing and heterogenous sources create the need for data curation. This paper outlines the strategies employed to bridge the gap between primary and secondary data sources. This involves the use of infrastructures and platforms with standardized data models and architectures that interface with each other seamlessly at the data storage and management layer and the use of ontologies at the semantic layer. Functionality and technical integration capacity tests are done to validate the technical effectiveness of the data storage and management layer and all testcases successfully pass and a comprehensive taxonomy is defined for the semantic layer. A meta(data) protocol is defined to guide data curation efforts and maturity model that employs a data curator log are enforced to serve as a tool for gauging the level of exchangeability and reusability of data. The outcomes demonstrate viability and reliability of the implemented data management framework to foster data collaboration.

Keywords: Interoperability, meta(data) curation, data management frameworks, standards, data models, data integration, primary data, secondary data

Table of Contents

1. Introduction

Interoperability refers to the ability of different institutions, systems, devices, and software applications to exchange, integrate, and interpret data effectively and accurately [1]. There are three main levels of interoperability namely, technical interoperability where the focus is on the protocol, connectivity, hardware and software issues that facilitate movement of data from point A to B [2], syntactic interoperability considers format and structure of the data and semantic interoperability goes deeper into terminologies, nomenclatures and ontologies [3].

According to FAIR (Findable, Accessible, Interoperable and Reusable) guiding principles for scientific data management and stewardship, interoperability is ensured by [4]:

- (Meta)data that use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- (Meta)data that uses vocabularies that follow FAIR principles.
- (Meta)data that includes qualified references to other (meta)data [5].

The advancement of interoperability is propelled by the need to answer a modelling question using isolated models that have no common native environment, and that raise significant data administration, alignment and transformation challenges [2]. The heterogeneous nature of biomedical data makes it even worse considering data coming in, in many forms, for example, genetic sequences, imaging studies, electronic health records, experiments, and clinical trial results, etc. In addition, these data types often have different standards, formats, and terminologies, making it difficult to integrate and analyze consistently [6].

Other problems that necessitate interoperability include the increasing complexity of scientific research [7], difference in schemas between different research applications, schema changes within these applications and platforms over time and individual research applications addressing a specific research process, leaving dynamic research processes in question. These can be solved using schema alignment, ensuring record linkage and is best done at the architecture design stage [2].

1.1 Background

Intervertebral disc degeneration is a common condition that affects the intervertebral discs in the spine, which are responsible for separating, cushioning, and articulating the vertebrae [8]. As these discs degenerate, they can become thinner and less able to perform their cushioning and articulating function, leading to chronic pain and disability [9]. Intervertebral disc degeneration is a significant cause of low back pain and contributes to global disability and healthcare costs [10].

Despite its prevalence [11], and the ever-increasing amounts of data about IVD, the sub-optimal conditions in which it is handled for instance being hidden in isolated silos and being handled by systems and formats that are proprietary and (or) restricted by intellectual property considerations, make it difficult to exchange, process and interpret IVD data. This impedes our understanding of the underlying causes and slows progress toward a better understanding of this condition and the development of more effective treatments[3].

Optimal interoperability is achieved when access and use of data and other digital objects is completely automated and is both human consumable and machine actionable [12]. This ensures that real world data can be used for large scale observational studies globally helping to address epidemiological questions and public health concerns, is presented in a manner that is ready for analysis and interpretation saving time by reducing pre-processing and data cleaning efforts, allows for remote collaboration of researchers and makes it possible for one analysis to be done across many different data sources leading to the generation of new medical insights[3].

In recent years, there has been a growing recognition of the importance of open research data and the potential thereof to accelerate scientific progress. In response, many funding agencies and research institutions have begun to adopt policies and initiatives to encourage research data curation. These efforts aim to make research data more interoperable, enabling it to be easily shared, combined, and analyzed with data from other sources [13]. This paper explores the interoperability of primary and secondary intervertebral disc degeneration data, intending to accelerate progress toward better

treatments and therapies. It focuses on the two main characteristics of interoperability which are the exchange and usability of data and information [2].

1.2 Methods

For the purposes of making data useful for synthesis efforts such as meta-analysis, verification of research results, parametrization of models as well as decision making [14], gold standard datasets undergo the cleaning, annotation, standardization and validation [4]. All these processes are encapsulated in meta(data) curation, whose steps include checking of files, understanding of the data, missing information checking and requesting, augmentation for findability, transformation of file formats, fairness evaluation and curation documentation[15].

1.2.1 Meta(data) Curation

Data curation is the intermediate step before data is stored. The preceding steps are data acquisition and data analysis [16]. Sharing data with a persistent identifier, linking publications with the persistently identified data and use of data repositories have been recommended as some of the best ways for caring for scientific data [17]. All these are part of the data curation process which is fully outlined below:

1. **Check files and metadata:** During this step, a dataset inventory is created and local appraisal is conducted, where the data is reviewed to make sure it is in the right repository and that all the metadata is provided by the researcher. The dataset file formats are identified and a working copy of the files is created. Key ethical considerations at this level include examining participant and data use agreements in order to determine the potential impacts of sharing the data.
2. **Understand and run data:** During this step, the datasets are examined more closely to make sure that the data is sufficient for any other researcher to understand and reuse. If it is not, then recommendations for additional documentation are done.
3. **Request missing information:** During this step, communication requests about any missing information in step 2 above are done.
4. **Augment metadata**: Once all the issues in the above step are addressed, then the metadata is standardized to conform to the repository specifications. Metadata models are then created at this level in order to link the different data points to

ontologies, to assist with machine interpretability. An example of the structure of the metadata model for a single field is as shown below:

```
{
 "FieldName": {
   "description": "A brief description of the field",
   "type": "The data type for the field (e.g., string, integer, boolean)",
   "ontologyTerm": "The ontology URI if applicable",
   "bioschemas": {
     "property": "The BioSchemas property that corresponds to the field",
     "marginality": "The level of importance of the field (minimum, recommended, or optional)",
     "cardinality": "The number of times the field can appear (one or many)",
     "bioschemasExample": "An example value or structure demonstrating the BioSchemas annotation"
   }
 }
}
```

5. **Transform file formats:** On this step, specialized formats and their restrictions are identified, with their opensource alternatives proposed. The original file formats are retained, but the opensource formats that are non-proprietary, preservation friendly and reusable are what is shared.
6. **Evaluate for Fairness:** Finally, to ascertain whether a dataset is gold standard, a final FAIR evaluation is conducted. For findability, a persistent digital identifier is checked, for accessibility, retrieval of the dataset via standard protocols is tested, for interoperability, metadata should be formatted in standard schema and for reusability, the data should be released with clear usage terms with clear indicators of who owns, and , stewards the data [18].

After the above data curation steps, the data is then stored in a repository. The APIs

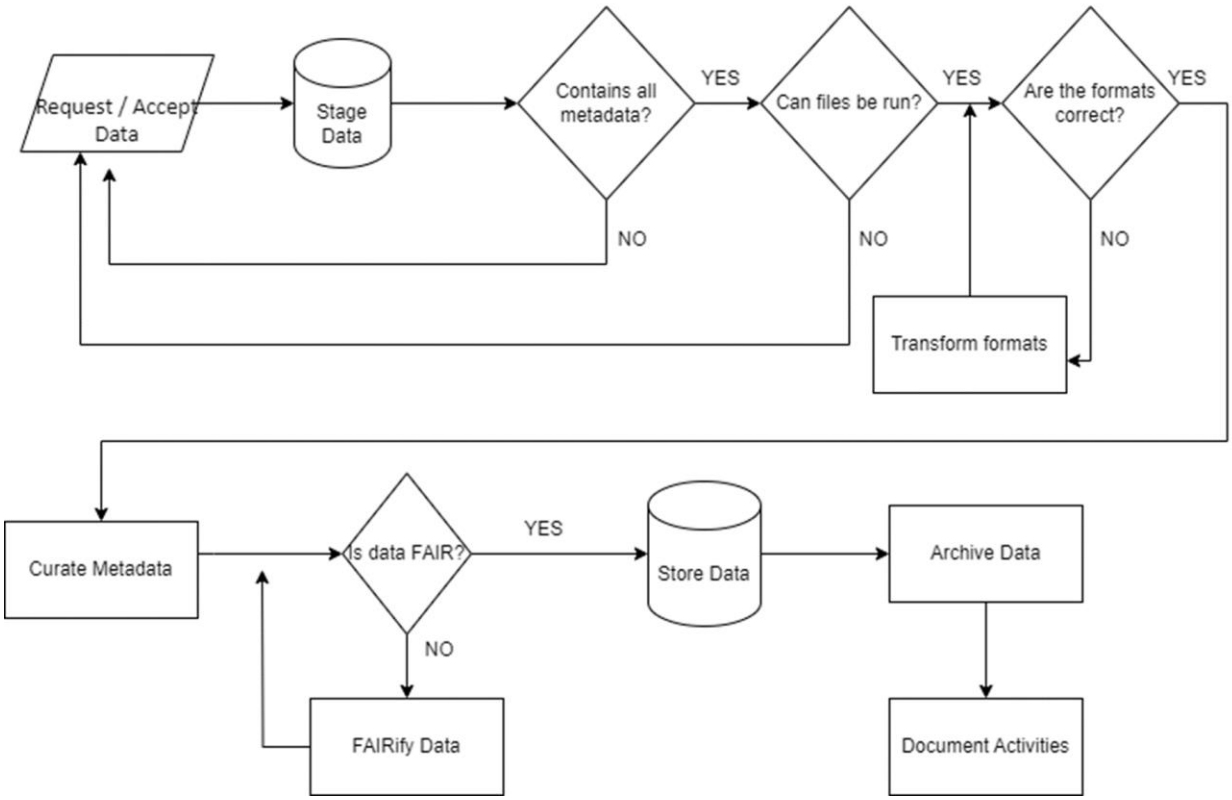

Figure 1 Meta(data) Curation Flow

(Application Programming Interfaces) of these repositories play a crucial role here, enabling automated data retrieval and submission, thus facilitating real-time data sharing and integration. In some cases there is use of several repositories. Effective use of APIs requires adherence to standardized protocols and security measures, ensuring reliable and secure data exchanges. This fulfils technical interoperability fully. To complete the syntactic and semantic interoperability, partially done through the data curation process, metadata curation is done, where the metadata is now made machine interpretable [19].

1.2.2 Meta(data) Curation Maturity Model

For the purposes of gauging how well Disc4All data is curated, we came up with a data curation maturity model, because there is no relevant data curation maturity model published scientifically. This model is a result of an analysis of several frameworks and models developed to guide data curation practices, working on their strengths and improving on their weaknesses. It contains five maturity levels:

1. Initial (Ad Hoc): At this foundational stage, data curation efforts are irregular and lack a structured approach. Formal policies or consistent methodologies for managing data are often absent, with success depending largely on individual initiative and makeshift solutions.
2. Managed (Repeatable): Organizations start to establish rudimentary policies and procedures for data curation. Although practices may not be uniformly applied, there's a growing recognition of the importance of consistency and the ability to replicate data management activities.
3. Defined (Standardized): By this phase, detailed data curation policies and procedures are well-documented and disseminated throughout the organization. Data curation follows a uniform method that is reliably executed across various projects.
4. Quantitatively Managed (Measured): At this level, data curation activities are both monitored and regulated. The organization implements specific, measurable objectives for data management quality, and tracks performance in relation to these objectives systematically.
5. Optimizing (Continuously Improving): This top maturity level is characterized by an ongoing commitment to enhancing data curation practices. Organizations consistently evaluate and update their data curation processes, integrating innovative tools and methods to boost efficiency and effectiveness.

The Data Curation Maturity Model above provides Disc4All with a structured framework to assess and enhance their data management practices systematically. By identifying different levels of maturity, from ad hoc efforts to continuous improvement, it helps organizations recognize their current position and guides them towards more effective and efficient data curation strategies. This model not only emphasizes the importance of establishing standardized procedures and measurable goals but also encourages a culture of continuous evaluation and adaptation. As a result, organizations can ensure the integrity, accessibility, and longevity of their data, which is essential for informed decision-making, compliance with regulations, and the advancement of knowledge in various fields.

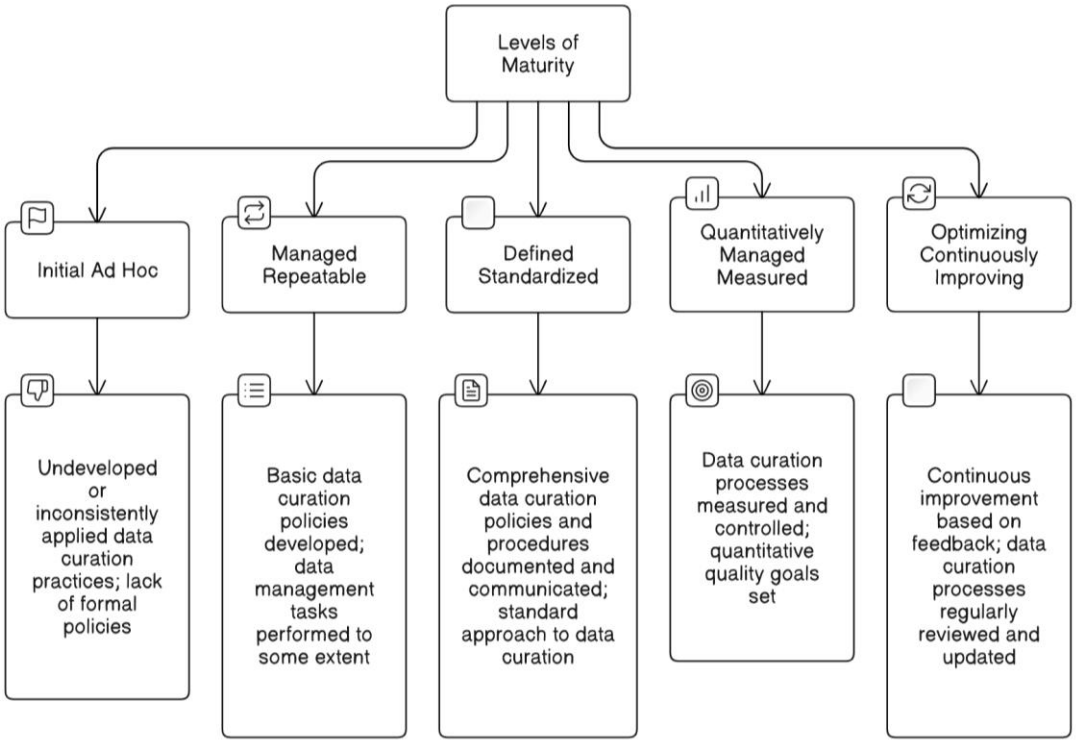

Figure 2 Data Curation Maturity Model

1.2.3 Data Curator Log

To operationalize the data curation maturity model, which encompasses technical interoperability as a key component of data curation excellence, a tool for scoring data curation activities was developed. It facilitates the benchmarking of data management practices and focuses on targeted improvements. It consists of several sections all aligned with the curation steps outlined earlier in this document. It then scores how well the different activities anticipated for each section are done and the cumulative score defines the level of maturity. Out of a total score of 66, the following are the bands for different maturity levels:

Level	Maturity Stage	Points Range
a)	Initial (Ad Hoc)	0-13
b)	Managed (Repeatable)	14-26
c)	Defined (Standardized)	27-39
d)	Quantitatively Managed (Measured)	40-52
e)	Optimizing (Continuously Improving)	53-66

Table 1 Data Curator Log Scores for Different Maturity Levels

2. The Data Commons

The Disc4All project uses a data Commons for storing and sharing data. This Data Commons is used for various purposes, including research, data analysis, and data management. It was built after analyzing the different data types within the project. This involved testing several open-source scientific data management components and identifying the most suitable. We then established policies and procedures using a data governance framework D 1.2, for accessing, processing, organizing and describing available data. Regardless of whether the data is primary or secondary, it is grouped into structured, semi-structured, and unstructured forms, all stored within different interconnected technologies that make it convenient to store and exchange information.

2.1 Structured data

Our framework of choice for structured data is Molgenis, an open source modular and extensible ecosystem that provides tools and services for data management, visualization, analysis, integration, and sharing. The framework is primarily used in genomics and bioinformatics and is designed to be easily customizable to meet the specific needs of different research projects and studies. Some of the critical features of MOLGENIS include a web-based user interface, data import and export capabilities, data visualization and exploration tools, and integration with other bioinformatics software and databases.

Structured data uploads to the internal Data Commons oblige the use of the Molgenis EMX format, which requires the data provider to have the data and metadata described in the same file before allowing them to import data to the project-specific database while still maintaining data integrity. After importing the data, the researcher still has the option to add, edit, delete, update or share data and metadata.

Some of the main advantages of using Molgenis are as shown below:

1. Scalability: The MOLGENIS framework is designed to handle large amounts of data and can scale to meet the needs of even the most extensive studies.
2. Modularity: The MOLGENIS framework is modular and extensible, allowing researchers to customize the platform to meet the specific needs of their projects.

3. Data Management: The MOLGENIS framework provides various tools for data management, including data import and export, validation, and quality control.
4. Data Visualization: The MOLGENIS framework provides a range of tools for data visualization, including data exploration and data reporting.
5. Data Analysis: The MOLGENIS framework provides a range of tools for data analysis, including statistical analysis, data mining, and machine learning.
6. User Management: The MOLGENIS framework provides tools for managing users and their access to the system, including user authentication, user authorization, and user roles.
7. Integration: The MOLGENIS framework is designed to integrate with other software and databases, making it easy to work with data from other sources.
8. Open source: The MOLGENIS framework is open source, which means it is freely available for anyone to use and modify. This also means that it has a large community of developers and users who can contribute to its development and provide support.
9. RESTful API: The MOLGENIS framework provides a RESTful API that allows users to interact with the system programmatically.

2.2 Semi-structured Data

Data resulting from simulated experiments do not conform to a strictly structured format. It is voluminous, hard to reproduce, and may need real-time processing. For these reasons, we chose an object-based storage model, where data is stored as a series of objects, each with a unique identifier and metadata. Open stack object storage, Swift, is not only the best candidate for this, but also the main option provided by Barcelona Supercomputing Center. Some of the features that make open stack the most suitable candidate for this include:

1. Scalability: OpenStack Object Storage is designed for horizontal scalability, meaning that it can handle large amounts of data and many users without performance degradation, which is essential for simulated biomedical data that can be voluminous.
2. Flexibility: Swift is built to handle unstructured and semi-structured data, which makes it a good choice for storing simulated biomedical data that may not

conform to a strict structured format, such as complex 3D models of organs or simulations of physiological processes.

3. Data durability: Swift provides redundancy by storing multiple copies of the data across different storage nodes, which can protect against data loss in the event of hardware failure. This is important for simulated biomedical data that can be valuable and hard to reproduce.
4. Multi-tenancy: Swift can handle multiple tenants, with separate authentication and authorization for each tenant, which is helpful for simulated biomedical data that may be sensitive or confidential.
5. API-driven: Swift can be accessed and managed via various APIs, allowing easy integration with other systems and software. This can be useful for simulated biomedical data that needs to be analyzed or processed in real-time, such as virtual surgical planning.
6. Cost-effective: Swift is open source, which means it can be deployed without incurring license costs. Additionally, Swift can be deployed on commodity hardware, which can be cost-effective for storing large amounts of data.

2.3 Unstructured data

Medical images are considered unstructured because they don't have a specific format that allows for easy searching, sorting, or analysis. However, the metadata and annotations associated with them can be semi-structured. They are typically stored in formats like DICOM or NIFTI, which include information about the image, but the actual image data is a matrix of pixels, which is unstructured and used mainly for visualization and interpretation by medical professionals. Our candidate for medical image storage and sharing is XNAT. It is the best candidate for medical image storage and sharing because of:

1. DICOM support: XNAT is designed to handle DICOM (Digital Imaging and Communications in Medicine) format, which is the standard file format for medical images, making it easy to import and export images from other imaging software and systems such as PACS (Picture Archiving and Communication Systems).
2. Data management: XNAT provides tools for organizing, managing, and annotating medical images, making it easy to search and retrieve specific images. It also allows for the creation of custom data types, which can be used

to store additional information such as patient demographics, clinical information and experimental data.

3. Data security: XNAT includes security features such as user authentication and access controls, which can be used to protect sensitive medical image data.
4. Data sharing: XNAT allows for easy sharing of medical images with other researchers and healthcare professionals, making it easy to collaborate on research projects and clinical studies.
5. Data processing: XNAT includes tools for processing medical images, such as image registration, segmentation, and visualization. It can also be integrated in other neuroimaging software such as FSL, SPM, and AFNI.
6. Multi-tenancy: XNAT can handle multiple tenants, with separate authentication and authorization for each tenant.
7. Cost-effective: XNAT is open source, which means it can be deployed without incurring license costs. Additionally, it can be deployed on commodity hardware, which can be cost-effective for storing large amounts of data.

2.4 Metadata

Metadata will be stored and shared in two different formats. The first will be through the use of a data catalogue by Molgenis and thereafter through a project specific ontology to facilitate semantic interoperability. Data-specific and project-specific details, including variable names, units of measurement, data collection methodologies, quality parameters, and project particulars like title, researchers and institutions involved, and study objectives are recorded.

More specifically, semi-structured data, stored in OpenStack Swift, adopts an object-based approach, with each object possessing a unique identifier. Descriptive metadata from semi-structured data delineates content and purpose, encompassing simulation type, model parameters, simulation outputs, and creator information. Unstructured data, managed via XNAT, contributes DICOM metadata for medical images, encompassing anonymized patient data, acquisition specifics, and image modality. XNAT-specific metadata includes user-defined tags, links to related data objects, and project affiliations.

Data-specific metadata is either embedded within the data itself (utilizing the Molgenis EMX format) or associated with objects (in the case of OpenStack Swift).
Secondly, descriptive metadata is captured through dedicated fields or annotations, facilitating context provision and enhancing data discovery and analysis.
This comprehensive metadata approach ensures that data within the Data Commons is well-documented, searchable, and adaptable for a variety of research endeavors.

3. Technical Interoperability

We reviewed the documentation to understand the functionality of the structured, semi-structured, and unstructured storages using their APIs, after identifying the key functionalities as creating, reading, updating, and deleting objects/entities. We then created test cases for each key functionality.

The test cases were then populated using the metadata below:

Test Case ID – Test Case Unique Identifier
Test Case Description – A description of what the endpoint achieves
Expected Result – Result expected if service is running correctly
Actual Result – Result obtained after running the test using postman CLI
Pass/Fail – Actual Result = Expected result

3.1 Structured data Technical Interoperability

The MOLGENIS framework supports various standards in the field of genomics and bioinformatics. Some of the main standards that are supported by the MOLGENIS framework include:

1 Data exchange standards: The MOLGENIS framework supports various data exchange standards such as VCF, BED, GTF, GFF, CSV, XLSX and FASTA. This allows for easy import and export of data from different sources and systems.
2 Data annotation standards: The MOLGENIS framework supports various data annotation standards, such as Gene Ontology (GO) and Human Phenotype Ontology (HPO), allowing easy data annotation and integration with other bioinformatics resources.

3 Data security standards: The MOLGENIS framework supports various data security standards, such as OAuth2 and OpenID Connect, allowing for easy integration with other security solutions and ensuring that the data stored in the MOLGENIS framework is secure.

4 It is also worth noting that the MOLGENIS framework is open source and actively developed, so the standards it supports may change over time.

3.1.1 Application Programming Interface

API collections and schemas for structured data within the Disc4All consortium: https://gitlab.bsc.es/disc4all/Structured-Interoperability-Tests

3.1.2 Test Cases

Running from Postman CLI:

```
stages:
  - automated-api-tests
automated-api-tests:
  stage: automated-api-tests
  image: cimg/base:2021.04
  before_script:
    # Installing Postman CLI
    - curl -o- "https://dl-cli.pstmn.io/install/linux64.sh" | sh
  script:
    # Login using your Postman API keys
    - postman login --with-api-key $POSTMAN_API_KEY
    - postman collection run "${CI_PROJECT_DIR}/postman/collections/XNAT REST API.json"
    # Lint your API using Postman CLI
    - postman api lint
```

Test results:

Test Case ID	Description	Expected result	Actual Result	Pass / Fail

1	Retrieve a list of entities using GET method	The API should return a list of entities in JSON format	The API returns a list of entities in JSON format	Pass
2	Create a new entity using POST method	The API should create a new entity and return a 201 status code	The API creates a new entity and returns a 201 status code	Pass
3	Update an existing entity using PUT method	The API should update the specified entity and return a 200 status code	The API updates the specified entity and returns a 200 status code	Pass
4	Delete an existing entity using DELETE method	The API should delete the specified entity and return a 204 status code	The API deletes the specified entity and returns a 204 status code	Pass
5	Interoperability with external system	The API should be able to receive and send data to an external system	The API is able to receive and send data to an external system	Pass
6	Retrieve a single entity using GET method	The API should return a single entity in JSON format based on the specified ID	The API returns a single entity in JSON format based on the specified ID	Pass
7	Retrieve a list of entities using GET method and filtering	The API should return a list of entities in JSON format based on the specified filtering criteria	The API returns a list of entities in JSON format based on the specified filtering criteria	Pass
8	Retrieve a list of entities using GET method and sorting	The API should return a list of entities in JSON format based on the specified sorting criteria	The API returns a list of entities in JSON format based on the specified sorting criteria	Pass
9	Retrieve a list of entities using GET method and pagination	The API should return a list of entities in JSON format based on the specified pagination criteria	The API returns a list of entities in JSON format based on the specified pagination criteria	Pass
10	File upload and download using POST and GET methods	The API should be able to upload and download files to and from the specified endpoint	The API is able to upload and download files to and from the specified endpoint	Pass

Table 2 Structured Data Interoperability Test Results

3.2 Semi-structured Data Technical Interoperability

XNAT supports various standards to ensure interoperability with other systems and facilitate data sharing. Some of the standards supported by XNAT include:

1. DICOM (Digital Imaging and Communications in Medicine): XNAT supports importing and exporting DICOM images and metadata, which is the standard format for medical imaging data.
2. NIFTI (Neuroimaging Informatics Technology Initiative): XNAT supports the NIFTI format, a popular format for storing and sharing neuroimaging data.
3. XCEDE (XML for Cognitive and Neuroimaging Data Exchange): XNAT supports XCEDE, which is a standard for describing and sharing cognitive and neuroimaging data.
4. MINC (Medical Imaging NetCDF): XNAT supports MINC, a format for medical images based on the NetCDF standard.
5. MIPAV (Medical Image Processing, Analysis, and Visualization): XNAT supports MIPAV, a software package for medical image processing, analysis, and visualization.

Overall, XNAT supports various standards to ensure that the data it stores and shares can be used by other systems and tools for further analysis and visualization.

3.2.1 Application Programming Interface

API collections and schemas for unstructured data within the Disc4All consortium: https://gitlab.bsc.es/disc4all/Unstructured-Interoperability-Tests

3.2.2 Test Cases

Running from Postman CLI:

```
image: atlassian/default-image:3
pipelines:
  default:
    - step:
        name: Run automated API tests using Postman CLI
        script:
          # Install Postman CLI
          - curl -o- "https://dl-cli.pstmn.io/install/linux64.sh" | sh
          # Login using your API Key
          - postman login --with-api-key $POSTMAN_API_KEY
          # Run your collection using Postman CLI
```

- postman collection run "${BITBUCKET_CLONE_DIR}/postman/collections/XNAT REST API.json"

Lint your API using Postman CLI

- postman api lint

Test Results:

Test Case ID	**Test Case**	**Expected Outcome**	**Actual Outcome**	**Pass/Fail**
1	Create Project	A new project is created and added to the XNAT system and return a 201 status code	A new project is created and added to the XNAT system and return a 201 status code	Pass
2	Read Project	A project is retrieved, and its details are displayed and return a 200 status code	A project is retrieved, and its details are displayed and return a 200 status code	Pass
3	Update Project	A project's details are updated, and the changes are saved and return a 200 status code	A project's details are updated, and the changes are saved and return a 200 status code	Pass
4	Delete Project	A project is deleted from the XNAT system and return a 204 status code	A project is deleted from the XNAT system and return a 204 status code	Pass
5	Upload Imaging Data	Imaging data is uploaded and stored in the XNAT system and return a 201 status code	Imaging data is uploaded and stored in the XNAT system and return a 201 status code	Pass
6	Download Imaging Data	Imaging data is downloaded from the XNAT system and return a 200 status code	Imaging data is downloaded from the XNAT system and return a 200 status code	Pass
7	Authentication	Users are able to authenticate and access the XNAT system and return a 200 status code	Users are able to authenticate and access the XNAT system and return a 200 status code	Pass
8	Authorization	Users are only able to access resources they are	Users are only able to access resources they are authorized to see and return a 200 status code	Pass

		authorized to see and return a 200 status code		

Table 3 Unstructured Interoperability Test Results

3.3 Unstructured Data Technical Interoperability

OpenStack Object Storage (also known as Swift) is an open-source software that provides scalable and redundant object storage services. It supports various standards and protocols to ensure interoperability with other systems and facilitate data access and management. Some of the standards and protocols supported by OpenStack Object Storage include:

1. S3 (Simple Storage Service): OpenStack Object Storage supports the S3 API, a widely used API for object storage supported by many other object storage systems, including Amazon S3.
2. Swift API: OpenStack Object Storage also provides its API, which is called the Swift API, for managing and accessing object storage.
3. OpenStack Swift API: OpenStack Object Storage supports the OpenStack Swift API, a widely used API for managing and accessing object storage in OpenStack environments.
4. HTTP/HTTPS: OpenStack Object Storage supports the HTTP/HTTPS protocol for accessing and managing object storage.

Overall, OpenStack Object Storage supports various standards and protocols to ensure that the data it stores can be accessed and managed by a wide range of systems and tools. This allows users to access their data from different clients and platforms.

3.2.1 Application Programming Interface

API collections and schemas for unstructured data within the Disc4All consortium:
https://documenter.getpostman.com/view/1730566/2s8ZDczfDD

3.3.2 Test Cases

Running from Postman CLI:

```
postman login --with-api-key {{postman-api-key-here}}
postman collection run 1730566-caa782c7-b27a-454e-afea-2fd64fd5d0db
```

Test Results:

Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
1	Create object	API returns HTTP 201 and object is created	API returns HTTP 201 and object is created	Pass
2	Retrieve object	API returns HTTP 200 and object is returned	API returns HTTP 200 and object is returned	Pass
3	Update object	API returns HTTP 202 and object is updated	API returns HTTP 202 and object is updated	Pass
4	Delete object	API returns HTTP 204 and object is deleted	API returns HTTP 204 and object is deleted	Pass
5	List objects in container	API returns HTTP 200 and list of objects is returned	API returns HTTP 200 and list of objects is returned	Pass
6	Retrieve object metadata	API returns HTTP 200 and object metadata is returned	API returns HTTP 200 and object metadata is returned	Pass
7	Update object metadata	API returns HTTP 202 and object metadata is updated	API returns HTTP 202 and object metadata is updated	Pass
8	Retrieve container metadata	API returns HTTP 200 and container metadata is returned	API returns HTTP 200 and container metadata is returned	Pass
9	Update container metadata	API returns HTTP 202 and container metadata is updated	API returns HTTP 202 and container metadata is updated	Pass
10	Delete container	API returns HTTP 204 and container is deleted	API returns HTTP 204 and container is deleted	Pass

Table 4 Semi Structured Interoperability Test Results

4. Syntactic Interoperability

Syntactic interoperability, crucial for seamless data exchange among various systems, primarily concerns the format and structure of data. In our project, the application programming interfaces (APIs) discussed earlier facilitate data provision in JSON (JavaScript Object Notation) and XML (Extensible Markup Language), chosen for their widespread adoption and comprehensive understanding across modern systems [20].

JSON and XML offer several advantages for achieving syntactic interoperability. Firstly, they are universally recognized standards for data exchange, ensuring that systems from diverse vendors and programming languages can effortlessly parse and comprehend the data structure [21]. Additionally, despite being machine-readable, both formats offer a degree of human readability, simplifying interpretation for developers and analysts and easing troubleshooting and debugging processes [20]. Moreover, their flexibility in representing complex data structures, including nested objects and arrays, is vital for modelling the rich data exchanges required within our project [20].

By adopting JSON and XML, our project achieves full syntactic interoperability, enabling seamless data exchange without the need to understand each other's internal data structures [22]. This streamlines integration efforts and ensures that data can be readily utilized by various components within our project environment.

5. Semantic Interoperability

Semantic interoperability is paramount for ensuring that different systems not only understand the format of exchanged data but also interpret its meaning consistently. This chapter explores two key approaches to achieving semantic interoperability within our project.

5.1 Semantic Annotation

The first approach involves annotating shared data using existing ontologies. Ontologies serve as formal representations of knowledge within specific domains, defining concepts, relationships, and attributes pertinent to those domains [23]. Leveraging existing ontologies related to the exchanged data provides several advantages. It establishes a common vocabulary and understanding of the data, reducing ambiguity and ensuring consistent interpretation across systems. Furthermore, it saves time and resources compared to developing new ontologies from scratch, and facilitates interoperability with external systems aligned with the same standards [24].

To demonstrate this look at the table below before it undergoes semantic annotation:

ID	Stimuli	Response	Relation

1	ADAMTS4/5	VEGF	ACTIVATOR
2	GDF5	ACAN	ACTIVATOR
3	GDF5	COL2A	ACTIVATOR
4	GDF5	IL-1β	INHIBITOR
5	GDF5	TNF	INHIBITOR
6	GDF5	MMP3	INHIBITOR
7	IFN-G	MMP13	ACTIVATOR

Figure 3 Data before annotation

Transformation document:

Column	Current Format	Better Format	Technique
ID	integer	integer	keep
Stimuli	string	application ontology IRI	MIREOT
Response	string	application ontology IRI	MIREOT
Relation	string	PATO IRI	MIREOT
Reference1	string	PMID	replace
Reference2	string	PMID	replace
Reference3	string	PMID	replace

Figure 4 Transformation strategy

and after annotation the table looks as shown below:

ID	Stimuli	Response	Relation
1	PR:Q6ZMM2/OGG:3000011096	OMIT:0036056	INO:0000017
2	OGG:3000008200	OGG:3000000176	INO:0000017
3	OGG:3000008200	OGG:3000001280	INO:0000017
4	OGG:3000008200	EFO:0003794	NCIT:C154898

5	OGG:3000008200	NCIT:C91692	NCIT:C154898
6	OGG:3000008200	OGG:3000004314	NCIT:C154898
7	PR:000000017	OGG:3000004322	INO:0000017

Figure 5 Data after annotation

5.2 Project Ontology

The second approach entails creating a project-specific ontology to address unique requirements not covered by existing ontologies. This custom ontology captures specific concepts, relationships, and attributes relevant to our project context, enhancing communication between developers and stakeholders and future-proofing the project for potential expansions.

By combining the use of existing ontologies with a project-specific ontology, our project establishes a robust framework for semantic interoperability [25]. This ensures that all systems involved maintain a coherent understanding of the exchanged data, facilitating accurate analysis and decision-making [26]. The appendix shows the taxonomy for the intervertebral disc ontology, based on project data.

6. Conclusion

Disc4All project data can be classified as structured, semi-structured, and unstructured. We established how XNAT, Molgenis, and the OpenStack Object Storage allow for the effective storage, retrieval, and visualization of the different data types, making them the best candidates for the creation of the project data storage and management layer. Furthermore, technical interoperability has been solved by availability of application programming interfaces of the different components making up the data commons, syntactic interoperability has been achieved by the format and structure of data exchange provided by the APIs and semantic interoperability is guaranteed by the use of ontologies to label submitted data and through defining of a project specific ontology as well.

4.1 Data Types and Solutions

The table below summarizes the different data types that have been identified inside the Disc4All project, their provenance, how we have categorized them, the solution addresses the problem of interoperability and which ontology class they belong.

Data Provenance	Data Type	Categorization	Solution	Relevant Ontology Components
Clinical Data	Patient Records	Structured	Molgenis	Phenotypes (IVD Degeneration, Modic Changes, AF Defects, Facet Tropism, Spine Alignment)
	MRI Scans	Unstructured	XNAT	Intervertebral Disc: Organ Culture Models: Whole-IVD: MRI
	Biopsychosocial Evaluations	Semi-Structured	Swift	Phenotypes
Molecular and Biological Data	Genomic Data	Structured	Molgenis	Intervertebral Disc: Experimental_Models: Cell_Biology: NP_Cells, CEP_Cells, AF_Cells: Transcriptomic_Measurements
	Proteomics and Metabolomics	Structured	Molgenis	Intervertebral Disc: Experimental_Models: Cell_Biology: NP_Cells, CEP_Cells, AF_Cells: Proteomics_Measurements
	Cytokine Profiles	Structured	Molgenis	Intervertebral Disc: Experimental_Models: Organ_Culture_Models: Whole-IVD: NP_Biology, AF_Biology, CEP_Biology: Proteomics_Measurements
	Microbiome Data	Semi-Structured	Swift	Intervertebral Disc: Experimental_Models: Computational_Biology: NP_Cell: Literature_Corpus

Experimental Data	Cell Culture Data	Structured	Molgenis	Intervertebral Disc: Experimental_Models: Cell_Biology
	Mechanical Load Data	Structured	Molgenis	Intervertebral Disc: Experimental_Models: Organ_Culture_Models: Whole-IVD: Mechanical Behaviour
	Nutritional and Biochemical Data	Structured	Molgenis	Intervertebral Disc: Experimental_Models: Organ_Culture_Models: Whole-IVD: Extracellular_Matrix_Composition
Computational and Simulation Data	In Silico Models	Semi-Structured	Swift	Intervertebral Disc: Computational_Models
	Simulation Outputs	Semi-Structured	Swift	Intervertebral Disc: Computational_Models: Finite_Element_Models
	AI-based Analyses	Structured	Swift	Intervertebral Disc: Computational_Models: Computational_Biology: Network_Models (Knowledge Based, Data Driven)
Epidemiological Data	Cohort Studies	Structured	Molgenis	Phenotypes: IVD Degeneration, Cohort-related studies
	Heritability Studies	Structured	Molgenis	Phenotypes: IVD Degeneration, Genetic studies

Table 5 Project data types and corresponding solutions and ontology class

4.2 Data Commons Architecture

The data interoperability effort must be placed in the larger context of the Disc4All platform. Researchers access the models and data through the VRE (virtual research environment). The data catalogue manages the metadata, while XNAT, SWIFT and Molgenis handle and show the datasets themselves, whether the data is structured, semi-structured or unstructured. The whole data pipeline is backed by the OpenStack Object

Storage for storage, retrieval and archival purposes and all data is integrated by the use of a project specific ontology.

The architecture just described is illustrated in the figure 6 , while the integrated primary and secondary data model is illustrated in figure 7 below.

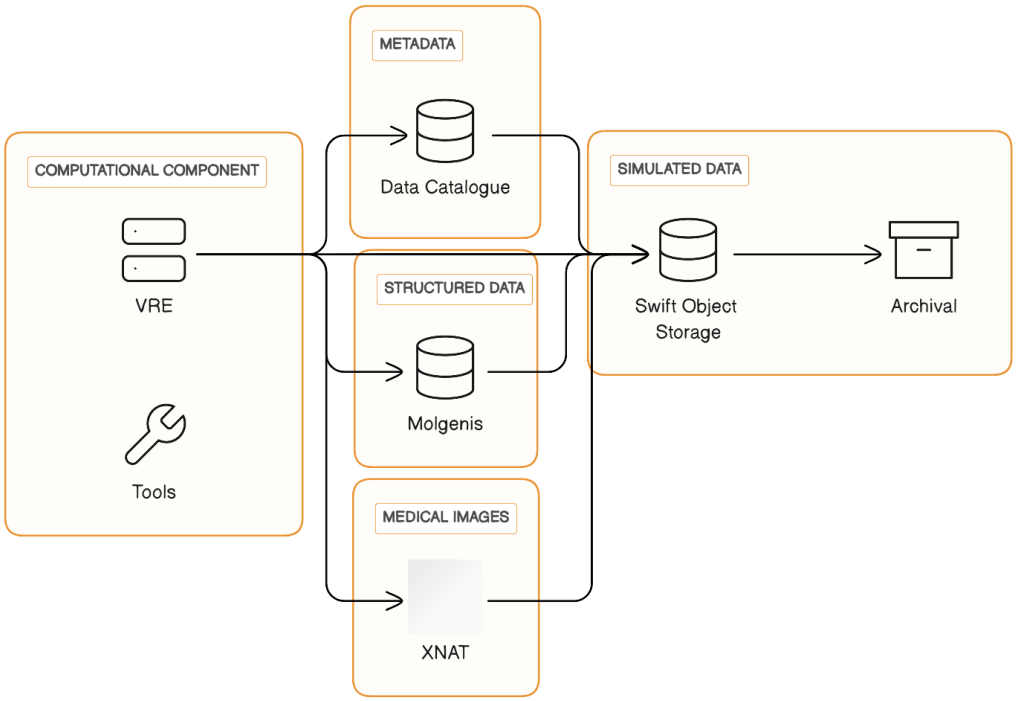

Figure 6 Data commons architecture

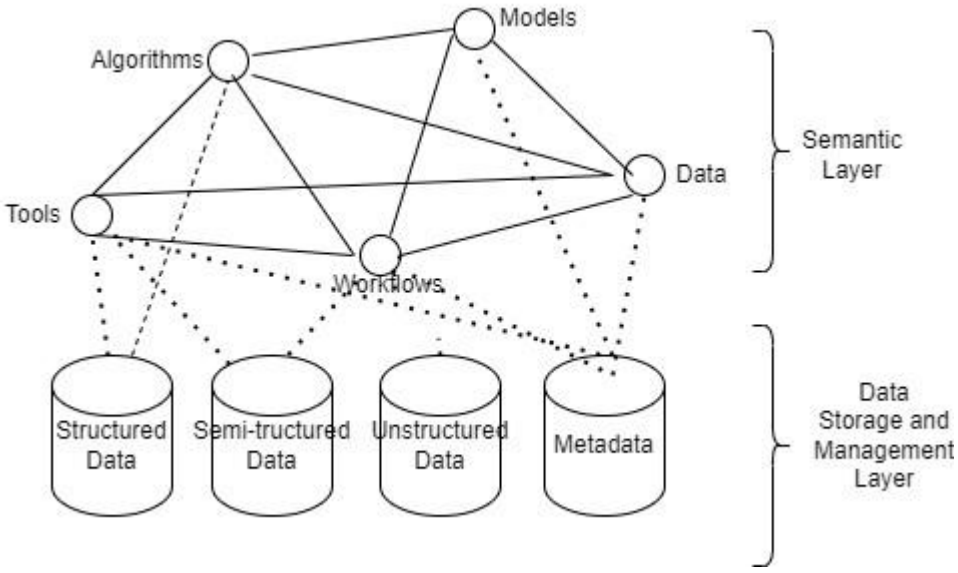

Figure 7 Integrated Primary and Secondary Data Model

5 References

[1] A. Valatavi, “Measuring Enterprise Application Software Interoperability Capability”.

[2] S. Y. Diallo, H. Herencia-Zapana, J. J. Padilla, and A. Tolk, “Understanding interoperability”.

[3] M. Lehne, J. Sass, A. Essenwanger, J. Schepers, and S. Thun, “Why digital medicine depends on interoperability,” *NPJ Digit Med*, vol. 2, p. 79, Aug. 2019, doi: 10.1038/s41746-019-0158-1.

[4] M. D. Wilkinson *et al.*, “The FAIR Guiding Principles for scientific data management and stewardship,” *Sci Data*, vol. 3, no. 1, p. 160018, Mar. 2016, doi: 10.1038/sdata.2016.18.

[5] A.-L. Lamprecht *et al.*, “Towards FAIR principles for research software,” *Data Science*, vol. 3, no. 1, pp. 37–59, Jan. 2020, doi: 10.3233/DS-190026.

[6] S. C. Davidson, “A Bigger Picture: Data Standards, Interoperability and Data Sharing,” *Spatial Database for GPS Wildlife Tracking Data*. Springer International Publishing, pp. 245–257, 2014. doi: 10.1007/978-3-319-03743-1_13.

[7] N. A. Kalinin and N. A. Skvortsov, “Difficulties of FAIR Principles Implementation in Cross-Domain Research Infrastructures,” *Lobachevskii J Math*, vol. 44, no. 1, pp. 147–156, Jan. 2023, doi: 10.1134/S199508022301016X.

[8] Y. Feng, B. Egan, and J. Wang, “Genetic factors in intervertebral disc degeneration,” *Genes & Diseases*, vol. 3, no. 3, pp. 178–185, Sep. 2016, doi: 10.1016/j.gendis.2016.04.005.

[9] J. P. Urban and S. Roberts, “Degeneration of the intervertebral disc,” *Arthritis Res Ther*, vol. 5, no. 3, p. 120, Mar. 2003, doi: 10.1186/ar629.

[10] A. Wu *et al.*, “Global low back pain prevalence and years lived with disability from 1990 to 2017: estimates from the Global Burden of Disease Study 2017,” *Ann Transl Med*, vol. 8, no. 6, pp. 299–299, Mar. 2020, doi: 10.21037/atm.2020.02.175.

[11] T. Vos *et al.*, “Years lived with disability (YLDs) for 1160 sequelae of 289 diseases and injuries 1990-2010: a systematic analysis for the Global Burden of Disease Study 2010,” *Lancet*, vol. 380, no. 9859, pp. 2163–2196, Dec. 2012, doi: 10.1016/S0140-6736(12)61729-2.

[12] S.-A. Sansone and P. Rocca-Serra, “Review: Interoperability standards,” p. 295232 Bytes, 2016, doi: 10.6084/M9.FIGSHARE.4055496.

[13] A. Zuiderwijk, R. Shinde, and W. Jeng, “What drives and inhibits researchers to share and use open research data? A systematic literature review to analyze factors influencing open research data adoption,” *PLoS ONE*, vol. 15, no. 9, p. e0239283, Sep. 2020, doi: 10.1371/journal.pone.0239283.

[14] W. K. Michener, “Ecological data sharing,” *Ecological Informatics*, vol. 29, pp. 33–44, Sep. 2015, doi: 10.1016/j.ecoinf.2015.06.010.

[15] "The DCN CURATE(D) Steps," Data Curation Network. Accessed: Mar. 05, 2024. [Online]. Available: https://datacurationnetwork.org/outputs/workflows/
[16] "All You Must Know About Data Curation in '23," AIMultiple: High Tech Use Cases & Tools to Grow Your Business. Accessed: Mar. 06, 2024. [Online]. Available: https://research.aimultiple.com/data-curation/
[17] A. Goodman *et al.*, "Ten Simple Rules for the Care and Feeding of Scientific Data," *PLOS Computational Biology*, vol. 10, no. 4, p. e1003542, Apr. 2014, doi: 10.1371/journal.pcbi.1003542.
[18] L. R. Johnston *et al.*, "Data Curation Network: A Cross-Institutional Staffing Model for Curating Research Data," *IJDC*, vol. 13, no. 1, pp. 125–140, Dec. 2018, doi: 10.2218/ijdc.v13i1.616.
[19] "(public web) Checklist for DCN Curators," Google Docs. Accessed: Mar. 06, 2024. [Online]. Available: https://docs.google.com/document/d/1RWt2obXOOeJRRFmVo9VAkl4h41cL33Zm5YYny3hbPZ8/edit?usp=embed_facebook
[20] H. Alani *et al.*, "Automatic ontology-based knowledge extraction from Web documents," *IEEE Intelligent Systems*, vol. 18, no. 1, pp. 14–21, Jan. 2003, doi: 10.1109/MIS.2003.1179189.
[21] "Extensible Markup Language (XML) 1.0 (Fifth Edition)." Accessed: Apr. 09, 2024. [Online]. Available: https://www.w3.org/TR/xml/
[22] T. Beale and S. Heard, "An ontology-based model of clinical information," *Stud Health Technol Inform*, vol. 129, no. Pt 1, pp. 760–764, 2007.
[23] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, Jun. 1993, doi: 10.1006/knac.1993.1008.
[24] D. Allemang and J. Hendler, *Semantic web for the working ontologist: modeling in RDF, RDFS and OWL*, Nachdr. Amsterdam Heidelberg: Elsevier, Morgan Kaufmann, 2010.
[25] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology".
[26] W. T. Adrian, N. Leone, and M. Manna, "Ontology-driven Information Extraction." arXiv, Dec. 18, 2015. doi: 10.48550/arXiv.1512.06034.

Appendix

Project Specific Ontology Taxonomy

Phenotypes

- IVD Degeneration
- Modic Changes
- AF Defects
- Facet Tropism
- Spine Alignment

- Sagittal
- Frontal

Intervertebral Disc

- Experimental_Models
  - Cell_Biology
    - NP_Cells
      - Transcriptomic_Measurements
      - Proteomics_Measurements
    - CEP_Cells
      - Transcriptomic_Measurements
      - Proteomics_Measurements
    - AF_Cells
      - Transcriptomic_Measurements
      - Proteomics_Measurements
  - Organ_Culture_Models
    - Whole-IVD
      - Mechanical Behaviour
        - Compression
        - Torsion
      - MRI
      - NP_Biology
        - Transcriptomic_Measurements
        - Proteomics_Measurements
        - Bio_Imaging
      - AF_Biology
        - Transcriptomic_Measurements
        - Proteomics_Measurements
        - Bio_Imaging
      - CEP_Biology
        - Transcriptomic_Measurements
        - Proteomics_Measurements
        - Bio_Imaging

- Extracellular_Matrix_Composition
  - NP
    - Proteoglycans
    - Collagen
    - Water
  - AF
    - Proteoglycans
    - Collagen
    - Water

Whole-Endplate

- Mechanical Behaviour
  - Fluid Permeation
- Micro-CT
- BioImaging
- CEP_Biology
  - Transcriptomic_Measurements
  - Proteomics_Measurements
- Extracellular_Matrix_Composition
  - CEP
    - Proteoglycans
    - Collagen
    - Water

Computational_Models

- Segmentation
  - Human
  - Bovine
- Finite_Element_Models
  - Organ_Models
    - Composition_Dependent
      - Mechanical
      - Mechano_Transport

- Porohyperelastic
  - Mechanical
  - Mechano_Transport
- Endplate_Models
  - Continuum
- Computational_Biology
- NP_Cell
  - Literature_Corpus
    - Soluble_Molecules
    - Genes
    - Regulation_Pathways
  - Network_Models
    - Knowledge_Based
    - Data_Driven
  - Agent_Based_Model